specification, it is a straight-forward matter to write the statements of the program executed by each "bubble". This may be done directly in the chosen programming language or by means of well-known intermediate steps such as structured English.

In terms of the above conventions, FIG. **4** is a context data flow diagram of the overall operating system for the video picture file portion of the video disk player. FIG. **5** is a level **1** child diagram expanding upon the file operating system "bubble" in FIG. **4** and showing the four modes of operation, that is, the "set-up", "edit", "normal" and "album modes". FIGS. **6** and **7** are level **2** child diagrams expanding upon two of the four modes show in FIG. **5**. FIGS. **8–10** are level **3** child diagrams expanding upon three of the separate editing levels shown by FIG. **7**. Table I (at the end of the specification) shows the data dictionary for the data names associated with the named vectors of FIGS. **4–10**.

The data flow diagrams of FIGS. **4–10**, in combination with the data dictionary listed in Table I, are self-explanatory to a programmer of ordinary skill in structured systems design. However, again for the general reader, some comments are helpful. Some of the process "bubbles" relate to circuit elements shown in FIGS. **2A** and **2B**. In the context diagram of FIG. **4**, a user control process **280** includes operation of the remote controller **24**, a video signal generation process **282** includes operation of the video circuit **205** and a graphics generation process **284** includes operation of the text generator **206**. The mechanical operating system **286** and the file operating system **288** include operating programs stored in the memory **201** and executed under direction of the microcomputer **200**. The picture track **290** and the magazine memory **292** correspond directly to operations involving the video tracks on the disk **10** and the remote memory **20** (FIG. **1**), respectively. (In the following description, data names will be printed in capital letters and refer directly to like names in the data flow diagrams and the data dictionary.)

With regard to FIG. **4**, USER SELECTIONS are received from the user control **280** via CONTROLLER INPUT to the mechanical operating system **286**. The file operating system **288** requests the current status of CONTROLLER INPUT via an INPUT REQUEST to the mechanical operating system **286**. The CONTROLLER INPUT is an 8-bit word indicating, by which bit is set, which button on the remote controller **24** is active (i.e., has been pressed). However, recalling the discussion of the "screens", not all buttons are active for a given screen. Therefore, the file operating system **288**, which controls the "screen" being displayed, masks the INPUT REQUEST. In other words, though INPUT REQUEST accommodates all 8 bits of CONTROLLER INPUT, it is responsive to a selected subset of bits depending on the "screen" in use. The response back to the file operating system **288** is MASKED INPUTS, indicating which bit passed by the mask is active. If a passed bit is active, the file operating system **288** performs whatever function corresponds to the button that was depressed.

The "menu screen (2)" is produced by a mode control process **294** shown in the level **2** diagram of FIG. **5**. The mode control process **294** triggers the four file operating modes previously described by appropriate SELECT signals, specifically by providing SET-UP SELECT to the set-up process **296**, EDIT SELECT to the edit process **298**, NORMAL SELECT to the normal process **300** and ALBUM SELECT to the album process **302**. Control is returned to the mode selection process **294** by NEW SELECTION, which is generated by pressing select when the user is pointing to "exit to menu" in any of the "edit" mode "screens" or by pressing the exit button. FIGS. **6** and **7** show exemplary sub-level data flow diagrams for the the set-up process **296** and the edit process **298**. The normal process **300** and the album process **302** are described in detail in related patent application (C) Ser. No. 644,166.

In the set-up process diagrammed by FIG. **6**, an initialization process **304** starts the set-up process. SET-UP SCREEN initializes the text signals associated with the "set-up screen", which are stored in a working memory **306** (such as the RAM **210** in FIG. **2B**). SET-UP READY causes the screen to print (print screen process **308**), drawing upon previous conditions stored in the magazine memory **292**. COMPLETE triggers the next input process **310**, where the system waits for the next MASKED INPUT. The input will to be to move the cursor (process **312**) by UP/DOWN or to make a selection (process **314**) by SELECT. When a selection is made, it is stored in the magazine (remote) memory **292** in place of the previous selection. The user gets out of this loop by NEW SELECTION, which leads back to the menu screen (2).

In the edit mode shown by FIG. **7**, editing is initiated by EDIT SELECT and the "edit option screen (3)" is then generated by the edit option control **316**. The editing process is partitioned into five levels as shown in FIG. **7**. The various SELECT commands determine which edit level is selected at any given time, as follows. DISK EDIT SELECT selects a first-level disk edit process **318**, PICTURE EDIT SELECT selects a second-level picture edit process **320**, ALBUM EDIT SELECT selects a third-level album edit process **322**, TEXT EDIT SELECT selects a fourth-level text edit process **324**, and RENAME EDIT SELECT selects a fifth-level rename edit process **326**. The first through third-level processes are respectively described in detail by FIGS. **8–10**. (The fourth and fifth-level processes are described in related patent application (C), Ser. No. 644,166.)

For the disk edit level shown by FIG. **8**, the DISK EDIT SELECT command turns control over to an initialization routine **328**, which generates the "disk select screen" (4) in process **330**. With DISK ID selected DISK EDIT READY is put out to a print screen process **332**. The latter process prints the "edit screen" (5), which displays the prerecorded picture along with a message identifying the display attributes, that is, the list of albums, which are associated with this level. The cursor is moved by sending UP/DOWN to process **334** and, once the cursor is adjacent a selected album, the decision to categorize this picture in the selected album is recorded via SELECT to an enterdelete process **336**. The decision is stored in the remote memory **20**. A next input process **338** looks for an active controller button by sending an INPUT REQUEST to the mechanical operating system **286** and waiting for a MASKED INPUT to come back. With the corresponding input active, the cursor is moved, a decision to enter or delete a picture from an album is made, the entire picture may be viewed (process **340**), or a new picture may be retrieved (by FWD/REV to a picture selection process **341**).

The picture edit level, described by FIG. **9**, has a data flow similar to the disk edit level shown by FIG. **8**. The "disk select screen" (4) and the "edit screen" (5) are